Structured-NeRF: Hierarchical Scene Graph with Neural Representation

Zhide Zhong[1,2*], Jiakai Cao[2*], Songen Gu[2,3], Sirui Xie[2,4], Liyi Luo[2], Hao Zhao[2], Guyue Zhou[2], Haoang Li[1], and Zike Yan[2†]

[1] HKUST (GZ) [2] AIR, Tsinghua University [3] Institute of Software, Chinese Academy of Sciences [4] Beijing University of Civil Engineering and Architecture

Abstract. We present Structured Neural Radiance Field (Structured-NeRF) for indoor scene representaion based on a novel hierarchical scene graph structure to organize the neural radiance field. Existing object-centric methods focus only on the inherent characteristics of objects, while overlooking the semantic and physical relationships between them. Our scene graph is adept at managing the complex real-world correlation between objects within a scene, enabling functionality beyond novel view synthesis, such as scene re-arrangement. Based on the hierarchical structure, we introduce the optimization strategy based on semantic and physical relationships, thus simplifying the operations involved in scene editing and ensuring both efficiency and accuracy. Moreover, we conduct shadow rendering on objects to further intensify the realism of the rendered images. Experimental results demonstrate our structured representation not only achieves state-of-the-art (SOTA) performance in object-level and scene-level rendering, but also advances downstream applications in union with LLM/VLM, such as automatic and instruction/image conditioned scene re-arrangement, thereby extending the NeRF to interactive editing conveniently and controllably.

Keywords: Hierarchical Scene Graph · Semantic and Physical Relationships · 3D Scene Editing

1 Introduction

Recent advancements in 3D scene reconstruction [30] and understanding [27] have been significantly propelled by the development of Neural Radiance Fields (NeRF) [2, 3, 24]. These models have demonstrated remarkable capabilities in rendering photorealistic scenes by encoding the volumetric density and color of a scene into a neural network. Traditional NeRFs primarily focus on capturing the visual appearance of scenes, which may not be sufficiently practical for domains such as robotics and augmented reality. We believe that an object-centric representation offers a more flexible approach to scene representation.

Existing methods of implicit learning of object semantic information usually output the semantic/instance channels with a shared MLP [4,43,46,48]. Despite

*Equal contribution. †Corresponding author.

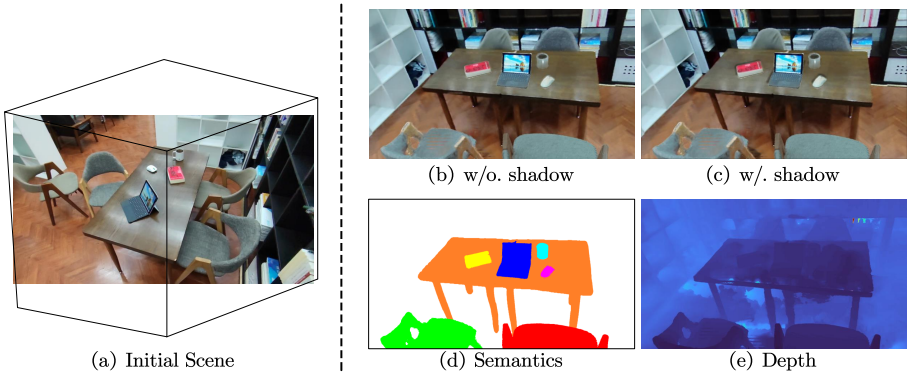


Fig. 1: Partial results of *office* scene from our datasets. By utilizing a freely catured video as input, we can decompose the scene into nodes and edges, and then render the rearranged scenes with shadow using volumetric rendering.

their effectiveness, they lack clear expression of spatial information about objects, and the globally shared weights also limit their scalability. Alternatively, some approaches parameterize objects as 3D boxes and treat each object as an independent neural model to learn separately [18,29,47], which facilitates downstream applications, such as scene editing. However, these methods only consider the properties of the objects themselves, without considering the semantic and physical relationships between objects within them. This limitation hinders their applicability in more complex scenarios where understanding and manipulating the relationships between objects are crucial, such as in interactive scene editing. To address these challenges, we introduce the Structured Neural Radiance Field (Structured-NeRF), a novel approach that leverages a hierarchical scene graph structure to represent indoor scenes comprehensively.

Structured-NeRF goes beyond the scope of existing object-centric methods by not only capturing the inherent characteristics of individual objects, but also by meticulously organizing the semantic and physical relationships between them. Existing NeRF editing methods [15,37] typically require meticulous adjustments by the user and may result in physically unrealistic phenomena, such as object clipping. However, the hierarchical scene graph structure facilitates a more organized and structured representation of the radiance field, enabling the model to manage the complex real-world correlations present in indoor scenes. By introducing optimization strategy based on these semantic and physical relationships, Structured-NeRF streamlines the process of scene arrangement, making it both efficient and accurate. In order to further enhance the realism of the rendered images, we have designed a method for rendering object shadows, inspired by shadow mapping techniques in computer graphics.

Our methodology not only achieves state-of-the-art (SOTA) performance in object-centric 3D scene reconstruction, but also enables interesting applications, including automatic and instruction/image-conditioned scene re-arrangements,

levering the powerful understanding capabilities of LLM/VLM [40, 50]. These advancements attest to the effectiveness, versatility, and scalability of this structure in scene understanding and reconstructing, as well as comprehending and editing complex real-world relationships.

In summary, the core contributions of our research are as follows.

• We introduce a novel method to organize NeRFs through a hierarchical scene graph, capturing both the inherent characteristics of objects and their semantic and physical relationships, enhancing scene understanding beyond traditional object-centric methods and achieved SOTA results in object-level and scene-level novel view synthesis.

• We introduce an innovative optimization strategy that leverages semantic and physical connections between objects to improve the efficiency and accuracy of scene editing.

• We introduce a method for rendering object shadows to increase the realism of the rendered images of new scenes, inspired by shadow mapping in graphics.

• We integrate the scene graphs with LLM/VLM, realising automatic and instruction/image-conditioned scene rearrangements. This has broadened the scope of NeRF's application in 3D scene editing.

2 Related Work

Object-centric Neural Radiance Field. Object-centric NeRF is crucial for applications such as autonomous driving and robotics, with a promising application being the construction of simulation platforms for embodied agents [49].

Implicit decomposition typically output the semantic/instance/feature channels with a shared MLP [4,43,46,48]. Semantic-NeRF [53] employs an additional semantic classification head to predict the semantic categories of sampled points, whereas [17] predicts the clip features of the sampled points, thus possessing the capability for open-set region querying. These methods effectively accomplish scene segmentation, but lack explicit spatial information due to their implicit representation, such as specific object location and size. Furthermore, the use of a globally shared network makes it challenging to isolate a particular object of interest, limiting their downstream applications, such as flexible scene editing.

In contrast to the implicit decomposition of the global neural radiance field, there are also explicit decomposition manners that decompose scenes into multiple car models [20,29]. Apart from the autonomous driving scenarios, the concurrent works of [11, 19] introduce object-decomposed NeRF-SLAM systems. Different object models are trained simultaneously to accelerate convergence and reduce the total parameters. These methods are effective at decoupling objects from scenes, but they only consider the intrinsic characteristics of the objects, overlooking the crucial semantic and physical relationships between them, which are vital for scene understanding and editing. We take a step forward and introduce a hierarchical scene graph structure and, based on this scene graph, meticulously design an automated object pose optimization pipeline, liberating users from the complexities of manual fine-tuning and editing processes.

NeRF Editing. Neural fields inherently encode shape and texture information of the scene, which implies considerable challenges in editing tasks that involve manipulating these fields. Instruct-Nerf2Nerf [12] introduces a method to iteratively edit the input images and optimize the underlying scene through re-trainng. However, its editing process is uncontrollable and does not allow for specifying and editing a local area [36]. Other methods [22, 25, 26] utilize the capabilities of the generation model [34] to perform content inpainting within the mask region, generating a new scene. Additionally, there are some methods use bounding box as a guide to insert 3D objects generated from text into the background radiance field [36, 37]. While they don't decompose the scene, thus unable to move objects that exist within the scene. These methods endow NeRF-editing with high controllability, yet they require users to meticulously adjust the object's pose or editing area to place object in a physically realistic position, significantly increasing the speed and complexity of the editing process. Compared to the aforementioned methods, our approach, by incorporating a scene graph as the organizational structure of the scene and undergoing semantic and physical optimization processes, can flexibly adjust the object's pose based on varying degrees of user input to automatically generate new scenes. This achieves a truly user-friendly scene editing functionality.

3 Method

As shown in Fig. 2, given a set of images with known poses, we initially utilize the Scene-decomposed NeRF method to decompose the scene and treat each object as a scene node. Subsequently, we use multimodal input to infer spatial relationships between object nodes. On the basis of the `support` relationships among the objects, we transform the nodes and edges into a hierarchical scene graph. This scene graph initially guides the **Semantic** relation optimization of the objects' poses from a top-down view. Afterwards, it utilizes the 'forces' between NeRFs to optimize the objects' placement to achieve a stable state that conforms to the **Physical** laws of the real world. Finally, we perform object-compositional rendering based on the scene graph to synthesize the new scene.

3.1 Structured Neural Radiance Field

Hierarchical Scene Graph. We used Open-Vocabulary 3D Hierarchical Scene Graph [6, 10, 33], in concert with neural radiance fields, as the scene representation, denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Here, $\mathcal{V} = \{v_i \mid i = 1, \ldots, N\}$ represents a set of object nodes which contains the properties and network weights of the objects. $\mathcal{E} = \{e_{ij} \mid i, j = 1, \ldots, N, i \neq j\}$ represents the semantic edges connecting each pair of objects. e_{ij} encompasses all spatial relationships $\left\{r^{i,j,k} = \left\{v^i, v^j, l^k\right\} \mid k = 0, \ldots\right\}$ between the nodes v_i and v_j. Note that we use the `support` relationship to determine the hierarchy of nodes, which determines the optimization order of their poses. For example, only after placing the cup holder, can the coffee cup be placed on top of it.

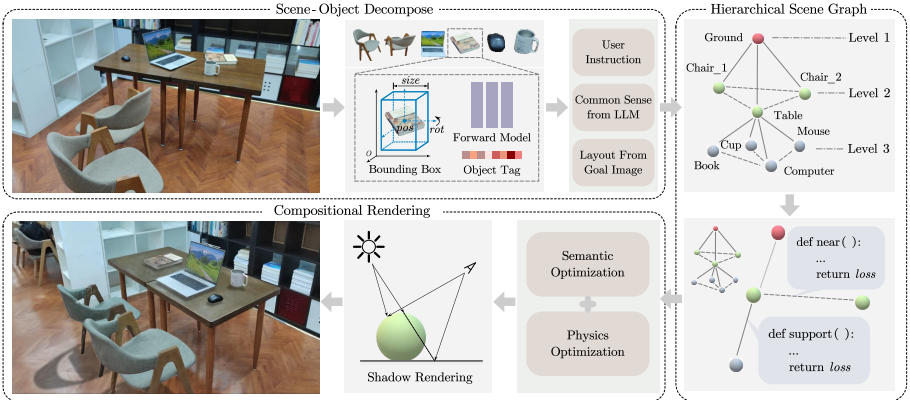


Fig. 2: Pipeline overview. Starting from a set of multi-view images, we first decompose the scene into independent nodes, followed by scene graph synthesis via LLM inference. Subsequently, we proceed through a semantic and physics optimization process, and render appropriate shadows for objects. Ultimately, the novel view of the new scene are synthesized.

Compositional Volume Rendering. The bottom left corner of Fig. 2 demonstrates our compositional rendering process. Given a camera pose $\mathcal{T}_i$, the ray $\mathbf{r} = \mathbf{o} + t\mathbf{d}$ emitted from the optical center $\mathbf{o}$ through a pixel can be determined. For the background node and all object nodes through which the ray emitted, we sample N points separately. That is to say, we can separate the whole ray into $m+1$ parts of $\mathbf{r} = \{\mathbf{r}^{\text{bg}}, \mathbf{r}^{\text{obj\_1}}, \cdots, \mathbf{r}^{\text{obj\_m}}\}$, where m is the number of intersected objects. As the intersection of the ray and the bounding box determines the starting and ending points $\{\mathbf{r}(t_{\text{in}}), \mathbf{r}(t_{\text{out}})\}$ of the ray through each object, points inside each object's bounding box are then transformed from the world space to their local canonical space accordingly.

The color and density values of the sampled points can be queried through forward passes given the associated forward models. By sorting the samples according to their depth values as $P_i \in \texttt{sorted}(\{P_i^{\text{bg}}, P_i^{\text{obj\_1}}, \cdots, P_i^{\text{obj\_m}}\})$, the rendering can be achieved through a composition of the standard volume rendering formula [23] as:

$$\hat{C}(\boldsymbol{r}) = \sum_{P_i} T_i \alpha_i c_i, \quad T_i = \exp(-\sum_{k=1}^{i-1} \sigma_k \delta_k), \tag{1}$$

where $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$, $\delta_i = t_{i+1} - t_i$.

Besides rendering the color of entire ray, we can also perform volume rendering separately for each node to generate the pixel color, depth and opacity as:

$$\hat{C}^k(\boldsymbol{r}^k) = \sum_{i=1}^{N} T_i^k \left(1 - \exp\left(-\sigma_i^k \delta_i^k\right)\right) \mathbf{c}_i^k, \tag{2}$$

$$\hat{D}^k(\boldsymbol{r}^k) = \sum_{i=1}^{N} T_i^k \left(1 - \exp\left(-\sigma_i^k \delta_i^k\right)\right) t_i^k, \tag{3}$$

$$\hat{O}^k(\boldsymbol{r}^k) = \sum_{i=1}^{N} T_i^k \left(1 - \exp\left(-\sigma_i^k \delta_i^k\right)\right). \tag{4}$$

where the background node is indexed as 0; $T_i^k = \exp\left(-\sum_{j=1}^{i-1} \sigma_i^k \delta_j^k\right)$ is the accumulated transmittance along the ray for the k-th object.

Optimization of Node Models. The photometric error minimization regarding the compositional rendering equals to the training of a single vanilla NeRF as:

$$\mathcal{L}_{\text{comp\_rgb}} = \left\| \hat{C}(\boldsymbol{r}) - C(\boldsymbol{r}) \right\|_2^2. \tag{5}$$

To enforce the gradient contribution to the right object node, we employ an object accumulation loss and an object color loss to supervise the associated object models. Given a binary mask $M^k(\boldsymbol{r}^k)$ that indicates if the ray first hits the surface of the associated object, the accumulation of each object should be consistent with the pixel-node association.

$$\mathcal{L}_{\text{obj\_acc}} = \sum_{k=1}^{K} \left\| \hat{O}^k(\boldsymbol{r}^k) - M^k(\boldsymbol{r}^k) \right\|_2^2, \mathcal{L}_{\text{obj\_rgb}} = \sum_{k=1}^{K} M^k(\boldsymbol{r}^k) \left\| \hat{C}^k(\boldsymbol{r}^k) - C(\boldsymbol{r}) \right\|_2^2. \tag{6}$$

We also use a pretrained inpainting model of LaMa [38] and LeftRefill [5] to predict the appearance of the occluded areas and supervise the color rendered by the background model. Considering LaMa is inherently a 2D model that lacks multi-view consistency, we adopt Omnidata [9] to predict the depth map of the inpainted image and serves as an auxiliary supervisory signal [51]:

$$\mathcal{L}_{\text{bg\_rgb}} = \left\| \hat{C}^0(\boldsymbol{r}) - C_{\text{inpaint}}(\boldsymbol{r}) \right\|_2^2, \mathcal{L}_{\text{bg\_depth}} = \left\| w\hat{D}^0(\boldsymbol{r}) + q - D_{\text{inpaint}}(\boldsymbol{r}) \right\|_2^2, \tag{7}$$

where w and q are learnable scale and shift factors that are optimized through training for scale and shift invariance.

The final loss for training our structured neural radiance field consists of five terms:

$$\mathcal{L} = \mathcal{L}_{\text{comp\_rgb}} + \lambda_1 \mathcal{L}_{\text{bg\_rgb}} + \lambda_2 \mathcal{L}_{\text{bg\_depth}} + \lambda_3 \mathcal{L}_{\text{obj\_acc}} + \lambda_4 \mathcal{L}_{\text{obj\_rgb}}, \tag{8}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are normalizing constants to balance the training.

3.2 Hierarchical Scene Graph Driven Editing

Building upon the structured representation outlined in Sec. 3.1, we exploit the powerful reasoning capabilities of LLMs to facilitate various editing applications,

the essence of which lies in leveraging foundation models to infer the relationships between objects. Here, we propose three levels of user input interfaces.

Automatic Scene Re-arrangement. We aim for scene re-arrangement to be automated, so we use the commonsense inherent in the GPT-4 [28] to infer the appropriate spatial relationships between the object nodes. For details, please refer to the supplementary material.

Instruction/Image-conditioned Re-arrangement. Sometimes, we aim to edit scenes based on user instructions. For instance, if a left-handed user specifies a relationship `(mouse, left to, laptop)`. For some application scenarios that required additional information, we suggest using a goal image to guide scene graph generation. We focus on extracting high-level semantic relationships, particularly the relative positions of objects, rather than replicating their exact locations from the image. We use GPT-4V [50] to recognize objects in the target image and describe their spatial interrelationships.

Once the relationships between the nodes are established, the system constructs a hierarchical scene graph based on the `support` relationships between the object nodes.

3.3 Object Pose Optimization

According to the guidance of the hierarchical scene graph, we optimize the object's pose layer by layer and the corresponding pseudo-code is given in supplementary material. Our optimization goal is to minimize loss, ensuring the positions of object nodes satisfy the constraints of all edges as much as possible, while also achieving physically plausible placement:

$$\mathcal{L} = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \left(\mathcal{L}_{\text{semantics}}(i,j) + \mathcal{L}_{\text{physics}}(i,j)\right) \tag{9}$$

where $\mathcal{L}_{\text{semantic}}(i,j)$ represents the constraint loss of the semantic edge between the i-th and j-th nodes, and $\mathcal{L}_{\text{physics}}(i,j)$ represents the constraint loss of the physical relationship between nodes.

Semantics Optimization. We map the difference between the object layout in the current scene and the object relationships defined in the scene graph to a set of loss functions that we interpret as the system's potential energy. Drawing from the principles of Langevin dynamics [13], we expect that as the optimization process unfolds, the spatial positions of the target objects will move in the direction where the system's potential energy decreases most rapidly (the negative gradient of potential energy with respect to position).

When traversing the scene graph from top to bottom in layers, we represent each node v_i by its 3D spatial coordinates p_i and the instance mask m_i rendered in the top-down view, as shown in Fig. 3. Taking node v_i as an example, we first identify each edge e_{ij} connected to it and the corresponding node v_j. Then, we map the triplet $(v_i, v_j, r^{i,j,k})$ to the loss function associated with the spatial-symbolic relationship. For details on the specific mapping function f, we refer

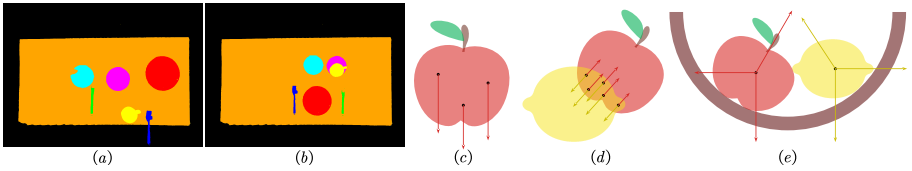


Fig. 3: An illustration of semantics optimization and physical optimization. (a) is the initial stage of the objects semantics on the table, while (b) is the optimized semantics. (c) and (d) visualize the forces that object may be effected, one is the gravity force while the other is the repulsive force, and (e) illustrates the stable stage that contacted objects would reach in the end.

the reader to the supplementary material.

$$\mathcal{L}\left(v_i, v_j, r^{i,j,k}\right) = f\left(p_i, m_i, p_j, m_j, r^{i,jk}\right) \tag{10}$$

$$\mathcal{L}_{\text{semantics}}(i,j) = \sum_{k}^{K} \mathcal{L}\left(v_i, v_j, r^{i,j,k}\right) \tag{11}$$

Physics Optimization. Relying solely on layout optimization to determine the position of objects is often not precise enough, since it can not simulate the physical interactions between objects. Since force is a condition that changes the state of motion of an object, a stable state is achieved when the resultant force is zero. Therefore, we introduce the Probabilistic Contact Model [21] to describe the contact relationship between objects:

$$\mathcal{L}_{physics}(i,j) = \boldsymbol{F}_{gravity} + \boldsymbol{F}_{repulsion} \tag{12}$$

where $\boldsymbol{F}_{gravity}$ represents the physical quantity depicting the mass of an object influenced by gravity, while $\boldsymbol{F}_{repulsion}$ is the repulsion force between two objects with overlapping volumes. Specifically, we have:

$$\boldsymbol{F}_{gravity} = m \cdot \boldsymbol{g};\ \ \boldsymbol{F}_{repulsion} = k \cdot \max(0, m_{overlap}) \cdot \boldsymbol{n}. \tag{13}$$

Here, m is the mass of the object, g is the gravitational constant, k is the elastic constant, while $m_{overlap}$ is the mass of overlapped volume, and $\boldsymbol{n}$ is the vector pointing from the overlapped volume toward the center of the object. To simplify the problem, we assume that objects are homogeneous materials, thus the mass is proportional to the volume, which allowing us to use the Monte Carlo method to estimate the mass of the object:

$$m_{obj} = \int_{x \in \mathbb{R}^3} \sigma\left(x\right) \approx \sum_{x \in X} \sigma\left(x\right). \tag{14}$$

Here, $\sigma(x)$ represents the density field of the object. Since $\sigma_i\left(x\right)$ represents the probability of object i occupying point x, the $\sigma_i\left(x\right)\sigma_j\left(x\right)$ is actually the

possibility that object i and j appears at the point x at the same time, which is not possible. Thus the repulsive force arises in the overlap of objects, pushing the two objects in opposite directions. We can calculate the mass of overlapped volumes as follows:

$$m_{overlap} \approx \frac{1}{N^{ij}} \sum_{x \in X^i \cap X^j} \sigma_i(x)\, \sigma_j(x) \tag{15}$$

where X^i and X^j are the sample points in object i and j respectively, and N^{ij} is the number of sample points that both in i and j. These forces will force objects to move in a small area until the overlapping area is minimized and balanced with the object's own gravity $\boldsymbol{F}_{gravity}$, as illustrated in Fig. 3.

3.4 Screen Space Shadow Map

In NeRF editing, a key element that greatly affects realism is the shadow of the object. We draw inspiration from the shadow mapping technique [45] in computer graphics to add high-fidelity shadow rendering to NeRF rendering. Whether a pixel in screen space will be affected by shadow is whether the light emitted from the light source is obstructed by other objects. Therefore, for each pixel point in screen space, we can determine the visibility of the world coordinate point corresponding to that pixel to the light source, implementing shadow mapping.

Let C_v be the RGB image rendered using composite rendering, and D_v be the corresponding depth image, while v_o and v_d are the starting point and direction of the ray, which can be calculated from the camera's position and viewing direction. After easily transformed these depth points into the light source's view, we can render rays from the light source L_o pointing to the depth points, and render depth that the light could travel before blocking by objects. By comparing the depth points in the light source's view, which is how far the light rays need to travel to illuminate that area, we can determine whether a pixel would be affected by the light.

4 Experiments

In this section, we introduce the dataset and provide evaluation results along with comparisons to baselines. Subsequently, we conduct an ablation study and analysis on the different components of the proposed method. Additional experimental result and analysis are available in the supplementary material.

4.1 Datasets and Implementation Details

We train our panoptic scene representation on multiple datasets. For quantitative evaluation, we follow the experimental setting of UDC-NeRF [43] and evaluate the proposed method on **ToyDesk** [48] and **ScanNet** [8] datasets. We also captured several lifelike indoor scenes using a handheld phone with resolution

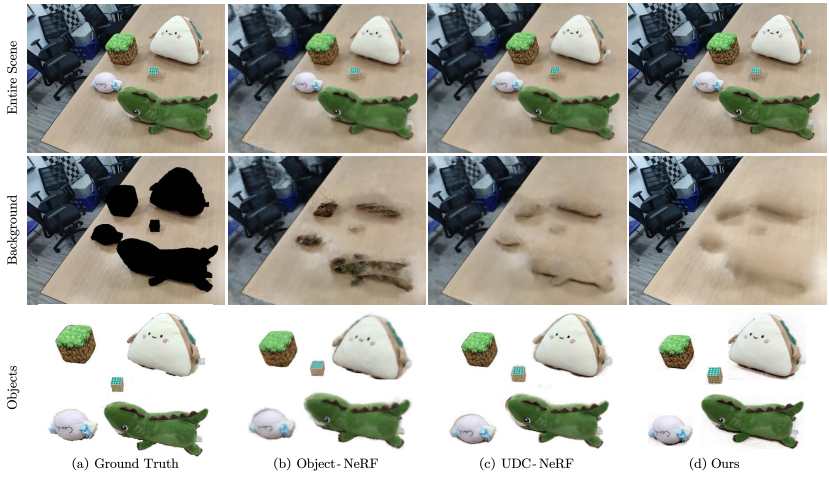


Fig. 4: Results of the decomposition on the ToyDesk [48]. Compared to Object-NeRF [48] and UDC-NeRF [43], our method has achieved superior results in rendering individual objects, backgrounds, and entire scenes.

Table 1: Quantitative comparison with other methods. The best results are shown in **bold**.

Scenes	Methods	PSNR ↑	SSIM ↑	LPIPS ↓
ToyDesk2	Object-NeRF [48]	24.815	0.7888	0.446
	UDC-NeRF [43]	25.756	0.8126	0.448
	Ours	**26.552**	**0.8507**	**0.186**
ScanNet	Object-NeRF [48]	25.264	0.8047	0.4094
	UDC-NeRF [43]	26.135	0.8249	0.395
	Ours	**30.360**	**0.8394**	**0.236**

1280×720 pixels for more qualitative analyses. Camera poses are calculated using COLMAP [35], while instance masks are estimated by XMem [7].

We take the default model (Nerfacto) of NeRFStudio [39] as our forward model architecture among the scene graphs. All experiments were conducted on a server with an AMD EPYC 7742 64-Core Processor and an NVIDIA GeForce RTX A4000 graphics card. Each model is trained for $30,000$ iterations using the Adam optimizer at a learning rate of $1e$-2. An exponential decay scheduler is applied, adjusting the learning rate to a final value of $1e$-4.

4.2 Baselines

Scene-level and Object-level Representation. We conduct experiments to evaluate the synthesized novel views of both foreground nodes and the back-

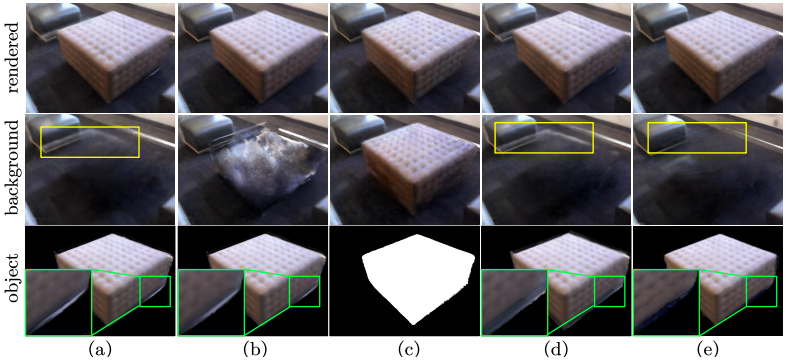


Fig. 5: Qualitative results of the ablation study: (a) w/o. 2D inpainted pseudo depth supervision; (b) w/o 2D inpainted pseudo RGBD supervision; (c) w/o object RGB supervision and 2D inpainted pseudo RGBD supervision; (d) w/o object accumulation supervision. (e) Ours.

Table 2: Ablation studies on the ScanNet dataset. Best shown in **bold** and the second-best shown underlined.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓
w/o Inpaint-Depth	30.68	0.8852	0.2127
w/o Inpaint-RGBD	<u>31.69</u>	**0.8897**	**0.2019**
w/o Obj. RGB and Inpaint-RGBD	31.31	0.8746	0.2182
w/o Obj. Acc.	30.99	0.8886	0.2110
Ours	**31.95**	<u>0.8895</u>	<u>0.2057</u>

ground node. The results are compared with the SOTA methods of **Object-NeRF** [48] and **UDC-NeRF** [43].

Scene Editing. For the downstream applications we propose, such as scene re-arrangement based on NeRF, there is currently no mature research. Therefore, after referencing DALL-E-Bot [16], we designed a baseline: **DALL-E-NeRF**, which interprets target objects and relationships as textual inputs and uses DALL·E 2 [32] to synthesize top-down view images and extract the layout of target objects. Subsequently, similar to our method, the images are synthesized by compositional rendering. In addition to the method based on NeRF, we also compare with an image manipulation method **Stable Diffusion** [34] to inpaint the given image from certain viewpoints. For controlled scene editing, we further input text instruction to Stable Diffusion to test its generative controllability.

4.3 Metrics

Rendering Quality. Following to [43], we measure the quality of the scene-level novel view synthesis on the evaluation metrics of **PSNR**, **SSIM**, and **LPIPS**

Fig. 6: Qualitative results of automatic re-arrangement. Compared to Stable Diffusion [34] and human arrangement, our method is capable of producing more reasonable renderings that hold true to real-world physical properties.

Table 3: Quantitative results using KID, FID and User Ratings.

	Dining Table			Kitchen			Office		
	KID ↓	FID ↓	USER ↑	KID ↓	FID ↓	USER ↑	KID ↓	FID ↓	USER ↑
Stable Diffusion [34]	0.29	267.92	1.69	0.33	298.40	1.81	0.05	155.95	1.90
DALL-E-NeRF	0.30	270.64	3.64	0.19	221.62	3.47	0.04	141.81	3.17
Ours	**0.26**	**251.92**	**3.96**	**0.14**	**197.22**	**3.85**	**0.04**	**138.55**	**3.88**

and render objects and backgrounds separately to demonstrate the effectiveness of the decomposition of our structured representation.

Scene Editing. We established two sets of evaluation metrics to comprehensively compare our method with the baselines. Firstly, inspired by recent work on indoor scene synthesis [31,41,42,44], we rendered the rearranged scenes using all methods from different viewpoints and calculated **FID** [14] and **KID** [52] compared to ground-truth scenes. Secondly, we conducted **user experiments**, with the rating criteria ranging from 1 (Incomplete) to 5 (Excellent).

4.4 Scene-level and Object-level Representation Ability

As illustrated in Fig. 4, the proposed method yields the best rendering results. For the background node, the black shadows are effectively removed while fine-grained details are recovered. In terms of object decomposition, our method provides more precise edges with clear details compared to other methods. The quantitative results in Tab. 1 further prove the effectiveness of our method.

4.5 Ablation Study

We also conducted comprehensive ablation studies on scene 0113_00 in the ScanNet dataset to validate the effectiveness of our loss terms. As demonstrated in Fig. 5 and Tab. 2, background inpainting can mitigate the decomposition

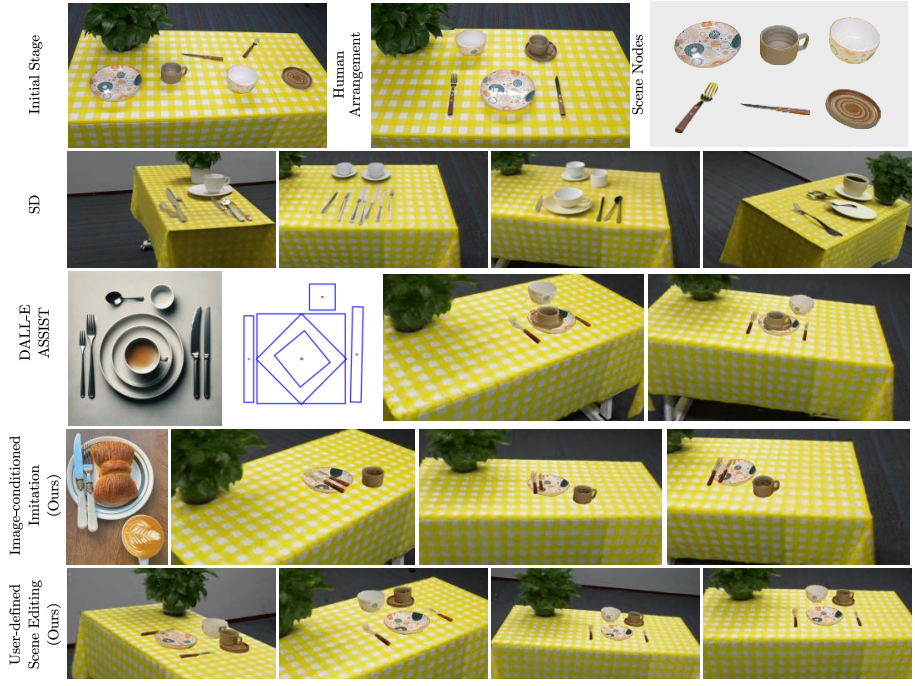


Fig. 7: Qualitative results of Instruction-conditioned rearrangement and Image-conditioned re-arrangement. Compared to Stable Diffusion [34], DALL-E-NeRF and human arrangement, our method is capable of generating new scenes that are better match user input.

ambiguity in the occluded area and generate reasonable observations within these areas. The accumulation loss results in cleaner areas in near-surface areas, which prevents rays that traverse the foreground bounding box without hitting an object are prevented from being used for rendering, thus avoiding gradient back-propagation to the wrong model.

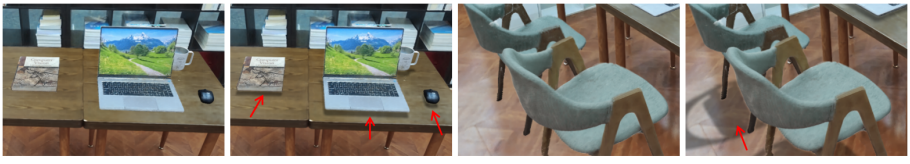
4.6 Scene Editing Applications

Automatic Scene Re-arrangement. Utilizing the object labels from the scene graph as input, we leverage the powerful zero-shot inferencing capabilities of GPT-4 [1] to deduce spatial relationships that align with human preferences. We depicted comparisons of rearranged scenes in Fig. 6. Our method, superior to Stable Diffusion, generates realistic and consistent images from various perspectives.

Instruction/Image-conditioned Re-arrangement. As shown in Fig. 7 and Tab. 3, our approach outperforms the baseline methods across all metrics and demonstrates high controllability. Understanding spatial and numerical relationships is typically challenging for methods based on generative models, yet our

(a) (b) (c) (d)

Fig. 8: Physical optimization for multi-object collision and support.



(a) (b) (c) (d)

Fig. 9: Qualitative comparison of whether shadow rendering is performed or not.

approach can satisfy user directives or target images more satisfactorily in most cases.

Physically Realistic Placement. We visualize the results of the physically realistic placement. In Fig. 8(a), the apple is added to the center and eventually stabilizes there. In Fig. 8(b), we add a lemon on the right side of the bowl, which collides with the apple, and both finally reach steady state, and the same occurs in Fig. 8(c). Upon adding the large orange last, due to the lack of remaining space at the bottom of the bowl, it slightly pushes the other objects outward and eventually stacks on top of the other fruits. The visual effects in the figures indicate that our method is consistent with the laws of physics.

Realistic Shadow Rendering. The comparison in Fig. 9 demonstrates that our method can significantly enhance the realism of the rendered image following editing.

5 Conclusion

In this paper, we introduce Structured-NeRF, a significant advancement in indoor scene representation. By introducing a novel hierarchical scene graph, we overcome the limitations of existing object-centric methods to scene editing by integrating the semantic and physical relationships between objects. This structure improves the efficiency, accuracy and realism of scene re-arrangement. Our experiments verify the superiority of Structured-NeRF in 3D scene reconstruction, understanding, and editing, demonstrating its potential to extend NeRFs to a broader range of application scenarios.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 13
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021) 1
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022) 1
4. Bing, W., Chen, L., Yang, B.: Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. In: ICLR (2023) 1, 3
5. Cao, C., Cai, Y., Dong, Q., Wang, Y., Fu, Y.: Leftrefill: Filling right canvas based on left reference through generalized text-to-image diffusion model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024) 6
6. Chang, H., Boyalakuntla, K., Lu, S., Cai, S., Jing, E., Keskar, S., Geng, S., Abbas, A., Zhou, L., Bekris, K., et al.: Context-aware entity grounding with open-vocabulary 3d scene graphs. arXiv preprint arXiv:2309.15940 (2023) 4
7. Cheng, H.K., Schwing, A.G.: Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In: ECCV. pp. 640–658. Springer (2022) 10
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR. pp. 5828–5839 (2017) 9
9. Eftekhar, A., Sax, A., Malik, J., Zamir, A.: Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In: ICCV. pp. 10786–10796 (2021) 6
10. Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K.M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., Gan, C., de Melo, C.M., Tenenbaum, J.B., Torralba, A., Shkurti, F., Paull, L.: Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning (2023) 4
11. Han, X., Liu, H., Ding, Y., Yang, L.: Ro-map: Real-time multi-object mapping with neural radiance fields (2023) 3
12. Haque, A., Tancik, M., Efros, A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: ICCV (2023) 4
13. Heller, G., Fetaya, E.: Can stochastic gradient langevin dynamics provide differential privacy for deep learning? (2023) 7
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017) 12
15. Jambon, C., Kerbl, B., Kopanas, G., Diolatzis, S., Drettakis, G., Leimkühler, T.: Nerfshop: Interactive editing of neural radiance fields. vol. 6 (2023) 2
16. Kapelyukh, I., Vosylius, V., Johns, E.: Dall-e-bot: Introducing web-scale diffusion models to robotics. IEEE Robotics and Automation Letters (2023) 11
17. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19729–19739 (2023) 3

18. Kong, X., Liu, S., Taher, M., Davison, A.J.: vmap: Vectorised object mapping for neural field slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 952–961 (2023) 2
19. Kong, X., Liu, S., Taher, M., Davison, A.J.: vmap: Vectorised object mapping for neural field slam. In: CVPR. pp. 952–961 (2023) 3
20. Kundu, A., Genova, K., Yin, X., Fathi, A., Pantofaru, C., Guibas, L.J., Tagliasacchi, A., Dellaert, F., Funkhouser, T.: Panoptic neural fields: A semantic object-aware neural scene representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12871–12881 (2022) 3
21. Le Cleac'h, S., Yu, H.X., Guo, M., Howell, T., Gao, R., Wu, J., Manchester, Z., Schwager, M.: Differentiable physics simulation of dynamics-augmented neural objects. IEEE Robotics and Automation Letters **8**(5), 2780–2787 (2023). https://doi.org/10.1109/LRA.2023.3257707 8
22. Liu, H.K., Shen, I., Chen, B.Y., et al.: Nerf-in: Free-form nerf inpainting with rgb-d priors. arXiv preprint arXiv:2206.04901 (2022) 4
23. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV. pp. 405–421. Springer (2020) 5
24. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 1
25. Mirzaei, A., Aumentado-Armstrong, T., Brubaker, M.A., Kelly, J., Levinshtein, A., Derpanis, K.G., Gilitschenski, I.: Reference-guided controllable inpainting of neural radiance fields. arXiv preprint arXiv:2304.09677 (2023) 4
26. Mirzaei, A., Aumentado-Armstrong, T., Derpanis, K.G., Kelly, J., Brubaker, M.A., Gilitschenski, I., Levinshtein, A.: Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20669–20679 (2023) 4
27. Nie, Y., Han, X., Guo, S., Zheng, Y., Chang, J., Zhang, J.J.: Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 55–64 (2020) 1
28. OpenAI: Gpt-4 technical report (2023) 7
29. Ost, J., Mannan, F., Thuerey, N., Knodt, J., Heide, F.: Neural scene graphs for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2856–2865 (2021) 2, 3
30. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 165–174 (2019) 1
31. Paschalidou, D., Kar, A., Shugrina, M., Kreis, K., Geiger, A., Fidler, S.: Atiss: Autoregressive transformers for indoor scene synthesis. Advances in Neural Information Processing Systems **34**, 12013–12026 (2021) 12
32. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents (2022) 11
33. Rana, K., Haviland, J., Garg, S., Abou-Chakra, J., Reid, I., Suenderhauf, N.: Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. arXiv preprint arXiv:2307.06135 (2023) 4
34. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF

conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 4, 11, 12, 13
35. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR. pp. 4104–4113 (2016) 10
36. Shahbazi, M., Claessens, L., Niemeyer, M., Collins, E., Tonioni, A., Van Gool, L., Tombari, F.: Inserf: Text-driven generative object insertion in neural 3d scenes. arXiv preprint arXiv:2401.05335 (2024) 4
37. Shum, K.C., Kim, J., Hua, B.S., Nguyen, D.T., Yeung, S.K.: Language-driven object fusion into neural radiance fields with pose-conditioned dataset updates (2023) 2, 4
38. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. pp. 2149–2159 (2022) 6
39. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., et al.: Nerfstudio: A modular framework for neural radiance field development. pp. 1–12 (2023) 10
40. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) 3
41. Wang, K., Savva, M., Chang, A.X., Ritchie, D.: Deep convolutional priors for indoor scene synthesis. ACM Transactions on Graphics (TOG) **37**(4), 1–14 (2018) 12
42. Wang, X., Yeshwanth, C., Nießner, M.: Sceneformer: Indoor scene generation with transformers. In: 2021 International Conference on 3D Vision (3DV). pp. 106–115. IEEE (2021) 12
43. Wang, Y., Wu, W., Xu, D.: Learning unified decompositional and compositional nerf for editable novel view synthesis. In: ICCV (2023) 1, 3, 9, 10, 11
44. Wei, Q.A., Ding, S., Park, J.J., Sajnani, R., Poulenard, A., Sridhar, S., Guibas, L.: Lego-net: Learning regular rearrangements of objects in rooms. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19037–19047 (2023) 12
45. Williams, L.: Casting curved shadows on curved surfaces. SIGGRAPH Comput. Graph. **12**(3), 270–274 (aug 1978) 9
46. Wu, Q., Liu, X., Chen, Y., Li, K., Zheng, C., Cai, J., Zheng, J.: Object-compositional neural implicit surfaces. In: ECCV. pp. 197–213. Springer (2022) 1, 3
47. Wu, Z., Liu, T., Luo, L., Zhong, Z., Chen, J., Xiao, H., Hou, C., Lou, H., Chen, Y., Yang, R., Huang, Y., Ye, X., Yan, Z., Shi, Y., Liao, Y., Zhao, H.: Mars: An instance-aware, modular and realistic simulator for autonomous driving. CICAI (2023) 2
48. Yang, B., Zhang, Y., Xu, Y., Li, Y., Zhou, H., Bao, H., Zhang, G., Cui, Z.: Learning object-compositional neural radiance field for editable scene rendering. In: ICCV. pp. 13779–13788 (2021) 1, 3, 9, 10, 11
49. Yang, Z., Chen, Y., Wang, J., Manivasagam, S., Ma, W.C., Yang, A.J., Urtasun, R.: Unisim: A neural closed-loop sensor simulator. In: CVPR. pp. 1389–1399 (2023) 3
50. Yang, Z., Li, L., Lin, K., Wang, J., Lin, C.C., Liu, Z., Wang, L.: The dawn of lmms: Preliminary explorations with gpt-4v (ision). arXiv preprint arXiv:2309.17421 **9**(1), 1 (2023) 3, 7

51. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. NeurIPS **35**, 25018–25032 (2022) 6
52. Zha, W., Li, X., Xing, Y., He, L., Li, D.: Reconstruction of shale image based on wasserstein generative adversarial networks with gradient penalty. Advances in Geo-Energy Research **4**(1), 107–114 (2020) 12
53. Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.J.: In-place scene labelling and understanding with implicit scene representation (2021) 3